SciNet: Lessons Learned from Building a Power-efficient Top-20 System and Data Centre

Chris Loken[1], Daniel Gruner[1], Leslie Groer[1], Richard Peltier[1], Neil Bunn[2], Michael Craig[2], Teresa Henriques[1], Jillian Dempsey[1], Ching-Hsing Yu[1], Joseph Chen[1], L. Jonathan Dursi[1], Jason Chong[1], Scott Northrup[1], Jaime Pinto[1], Neil Knecht[1], Ramses van Zon[1]

[1] SciNet HPC Consortium, University of Toronto, 256 McCaul St, Toronto, ON, M5T 1W5, Canada

[2] IBM Canada Ltd., 3600 Steeles Ave. E, Markham, ON, L3R 9Z7, Canada

E-mail: info@scinet.utoronto.ca

Abstract. SciNet, one of seven regional HPC consortia operating under the Compute Canada umbrella, runs Canada's first and third fastest computers (as of June 2010) in a state-of-the-art, highly energy-efficient datacentre with a Power Usage Effectiveness (PUE) design-point of 1.16. Power efficiency, computational "bang for the buck" and system capability for a handful of flagship science projects were important criteria in choosing the nature of the computers and the data centre itself. Here we outline some of the lessons learned in putting together the systems and the data centre that hosts Canada's fastest computer to date.

1. Introduction — The Current State of Affairs

The key compute systems at SciNet were installed beginning in November 2008 and include two clusters along with a shared 1.5 PB storage system which is visible to all 3,900 compute servers. The 30,000 core General Purpose Cluster (GPC) with a peak theoretical speed of 306 TFLOPS and the 3,300 core Tightly-coupled Capability System (TCS) with a peak theoretical of 60 TFLOPS are all housed in a new green datacentre with a measured average annual PUE of less than 1.2. These two clusters ranked numbers 53 and 16 in the world, respectively, when they first appeared on top500.org (in November 2008 and June 2009).

All systems were fully online and available to the entire Canadian research community as of August 2009. Uptake in terms of system utilization as well as numbers of users and Principal Investigators (PIs) was very quick, as shown in Figure 1. The speed at which the GPC reached 90%+ utilization, particularly since this roughly doubled the number of compute cycles available to Compute Canada researchers nationally, is remarkable. The more specialized TCS system took longer to reach this level of utilization, and because the number of users is smaller, usage is noticeably burstier. A total of more than 200 million CPU-hours were used in the first full year for research projects including cleaner-burning combustion engines, the effect of global warming on the Greenland ice-sheet, regional climate change in Ontario, the role of mantle convection in the formation of supercontinents on Earth, understanding the molecular basis for a substance which appears to eliminate Alzheimer's symptoms in mice, the molecular forces responsible for the remarkable properties of elastin and the body's defence mechanisms, the design of

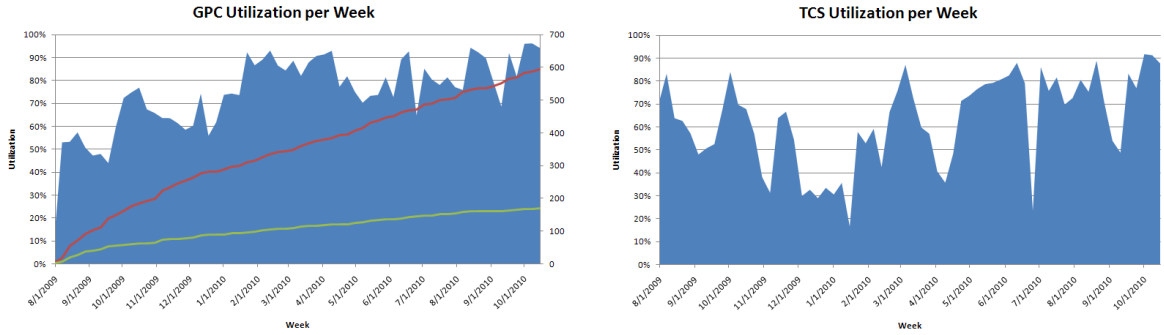

Figure 1. Utilization of the General Purpose Cluster (left) shown as shaded area, with number of users and PIs plotted as red and green lines respectively, over the first year of the SciNet systems being publicly available; shown on the right is the utilization data for the TCS.

low-emission aircraft, and the magnetic fields of galaxy clusters. After a year of such work, important SciNet-powered publications in astrophysics [1–17], space science [18], combustion research [19–25], and chemistry [26], are starting to appear in a variety of journals. The ATLAS experiment, that started taking high-energy proton-proton collision data at the LHC in 2009, also has publications that have depended on the analysis facility and crucial simulation data sets that were produced at SciNet[27–30].

SciNet is a consortium of the University of Toronto and its affiliated research hospitals including Baycrest Centre for Geriatric Care, Bloorview MacMillan Children's Centre, Centre for Addiction and Mental Health, Hospital for Sick Children, Mount Sinai Hospital, Ontario Institute for Cancer Research, St. Michael's Hospital, Sunnybrook Health Sciences Centre, University Health Network and Women's College Hospital. Funding comes from the federal government (Canada Foundation for Innovation and the Natural Sciences and Engineering Research Council), provincial government (Ontario Ministry for Research and Innovation and the Ontario Research Fund – Research Excellence) as well as the University of Toronto (the faculties of Arts & Science, Engineering, Medicine and Scarborough).

SciNet is one of the seven regional High Performance Computing (HPC) consortia which make up Compute Canada, an umbrella organization leading the creation of a powerful national HPC platform for research. This national platform integrates HPC resources at seven partner consortia across the country to create a dynamic computational resource. Compute Canada integrates high-performance computers, data resources and tools, and academic research facilities around the country. These integrated resources represent close to a petaFLOPS of computing capability, in addition to substantial online and long term storage with rapid access and retrieval over Canada's national, provincial and territorial high-performance networks.

Working in collaboration, Compute Canada and the university-based regional HPC consortia provide for overall architecture and planning, software integration, operations and management, and coordination of user support for the national HPC platform. As a national organization, Compute Canada coordinates and promotes the use of HPC in Canadian research and works to ensure that Canadian researchers have the computational facilities and expert services necessary to advance scientific knowledge and innovation.

SciNet has a modestly sized support team of 5 system administrators and 4 technical analysts, plus two Chief Technical Officers and a Scientific Director. Nine of the technical staff have science PhDs with a research background in HPC which naturally facilitates interactions with users.

Under the guidance of the CTO, the system administrators are tasked with the challenge of keeping Canada's largest computers running, optimizing system performance, keeping hardware

and firmware updated and interacting with vendors to deal with hardware issues.

The technical analyst team, led by the CTO-software, have as their primary responsibility to work with the users to enable their applications to be run at SciNet scales. Another major component of the software team's responsibilities are the tutorials and courses given at SciNet[1]. The analysts also install and maintain software libraries and applications relevant to the users. The size of this group should be constrasted with the computationally very similar HECToR group in the UK which employs 20 full time analysts solely for working with users to develop their codes (*i.e.*, not including "front line support" for compilation issues and the like), or with US laboratories like ORNL where the rule of thumb is "one FTE per three supported projects" (see for instance [31]).

The support team furthermore helps users with a wide range of questions and problems. Apart from login and access problems, these also include questions on how to use SciNet's systems, how to use the systems most efficiently (including how to parallelize and optimize their code), and how to use and/or install libraries. There is no formal ticketing system, but a user can initiate a request for support by emailing a central support list, which all members of the group receive. This informal system remains practical because the group is small and located in the same office, and furthermore allows for a quick response time. The most efficient way of interacting and helping users, however, is through face-to-face meetings, and therefore users are often invited to schedule a meeting with an analyst. Unfortunately, this is not always an option for remote users in which case video links have been successfully used.

2. History

The development of SciNet at the University of Toronto stretches back to 1999 with the timing of an award of $7.4M from the Canada Foundation for Innovation (CFI) and the Province of Ontario for the founding of PSciNet, an acronym which then stood for the Physical Sciences computing NETwork. This funding was received in response to an application prepared by a group consisting of astrophysicists from the Canadian Institute for Theoretical Astrophysics, chemical physicists from the Department of Chemistry and planetary physicists from the Department of Physics. The funds were invested to acquire three distinct computer systems, each one designed to serve the special needs of each of these collaborating groups and to be operated by them as separate systems.

A second proposal to CFI for further PSciNet development was funded in 2003 for $11.2M, this time to a group consisting of high energy experimental particle physicists from the Department of Physics, planetary physicists from the Department of Physics and a third group consisting of aerospace and mechanical engineers from the University of Toronto Institute for Aerospace Studies (UTIAS) and from the Department of Mechanical and Industrial Engineering. This funding was employed to acquire two new cluster systems as well as an upgrade to the vector system employed by the planetary physics group.

The SciNet consortium, with the "P" for "Physical" dropped from the acronym, was established in 2005 through the continuing collaboration of all five of these previously involved groups, joined by their colleagues from the areas of computational biology, genomics and bioinformatics at both the University of Toronto and the ten research hospitals affiliated with it. SciNet participated in developing the Compute Canada response to the 2006 National Platform Fund (NPF) call for proposal from CFI and was allocated $15M of the total amount awarded to Compute Canada in December 2006. The CFI award was matched by the Ontario provincial government and supplemented by the University of Toronto in order to provide SciNet with a total capital budget of $32.8M.

[1] Upcoming courses are listed at `https://support.scinet.utoronto.ca/courses/`, and course material is generally posted on the support wiki, `https://support.scinet.utoronto.ca/wiki/index.php/SciNet_User_Support_Library`

Complications in administering the NPF award across 7 consortia and more than 15 institutions delayed the issue of the SciNet Request for Proposals (RFP) until Jan 2008. After reviewing all proposals a final contract with IBM was signed in July 2008 for the construction of the datacentre (in an existing building) and the installation of two clusters and storage. Renovations began in late Aug 2008, the first cluster (the TCS) and storage system were installed in November and opened to friendly users in December, the datacentre was fully completed in February 2009 and the installation of the largest cluster (the GPC) began in March with the arrival of the first IBM iDataPlex servers based on the brand-new Intel Nehalem CPU architecture. Friendly user period for the GPC began in May and both systems were fully opened to researchers from across Canada at the beginning of August 2009.

3. Planning and Acquisition

Beginning even before the NPF Call for Proposal, the SciNet Technical Advisory Committee (STAC) actively engaged in planning and preparation for a future HPC system(s) by conducting user surveys, studying existing HPC facilities at University of Toronto and other sites in North America and consulting with industry concerning datacentre design as well as HPC equipment futures and trends. By the time the NPF proposal was submitted in mid-2006, it was clear that local researchers required significant capacity and capability computing and that the wide-range of research needs (from high-resolution global climate modeling to high-throughput data analysis for ATLAS and various Cosmic Microwave Background experiments) would likely require two systems with different compute node characteristics and network architectures. The requirement of some workflows to utilize both compute systems and a desire to use storage space as efficiently as possible drove the consensus towards a common, shared storage system.

During the planning process, it also became clear that the total power, cooling and space requirements being contemplated were far in excess of what was available in existing datacentres on campus. Extensive searches for appropriate space on and off campus and consultations with various co-location facilities led to the conclusion that SciNet would have to build or renovate a new datacentre (though solutions based on shipping containers would be considered). An off-campus, single storey, slab-on-grade "brownfield" site with adequate space and power was identified in 2007 and rented in 2008 ready to be converted into a datacentre.

After much deliberation, the STAC concluded that the SciNet requirements for two compute systems, storage and a datacentre coupled with the constraints of fixed capital and operating budgets gave rise to an optimization problem for system size and datacentre cooling design that was likely best solved by vendors. For example, the capability system was likely to be more expensive per Flop and to generate more heat per Flop than the capacity system though the dollar differentials depended on exact system type and configuration. In addition, possible compute system candidates required different cooling system designs (e.g. air or water-cooled systems) with some of the air-cooled versions requiring significantly greater airflows than normal in standard raised-floor datacentres. It was therefore decided to issue a single RFP requesting four components: datacentre, storage and two compute systems. Vendors were given a total capital budget and, in addition, were required to provide a 5-year operating budget (for power and all hardware and infrastructure maintenance) that fit within the SciNet operating budget. SciNet reserved the right to "cherry-pick" the various components from different responses.

The *de novo* nature of putting together the SciNet datacentre, two compute systems, and storage, allowed vendors a great deal of flexibility; but there were strong restrictions on the resulting solution – there were fixed capital and 5-year operating budgets dominated by power costs, but within that envelope SciNet obviously wanted the best possible systems for its mix of problems which included both throughput computing and tightly coupled, massively parallel jobs. The SciNet RFP, rather uniquely, gave the vendors all four components in the bid and maximized their flexibility in designing and pricing the entire package.

Table 1. GPC RFP tentative minimum specifications

	serial	parallel	shared-memory
Peak theoretical Flops	$>$ 300 Tflops	$>$ 100 Tflops	n.a.
RAM	$\geq$ 2 GB per core	$\geq$ 2 GB per core	128 and 256 GB per node
Total RAM	60 TB	20 TB	n.a.
Largest non-blocking single MPI job	256 cores	$\geq$ 5,000 cores	8-16 cores
Network connection	$\geq$ 100 MB/s/core	$\geq$ 1 Gb/s/core	100 MB/s/core
Network latency (node to-node)	$<$ 60 μs	$<$ 5 μs	$<$ 5 μs
I/O to disk (sequential, large blocks)	$\geq$ 1 MB/s/core	$\geq$ 2 MB/s/core	$\geq$ 2 MB/s/core
number of simultaneous I/O streams	10,000	5,000	100

SciNet presented participating vendors with a challenging problem of optimization. The overall solution was constrained by a single point-in-time funding model which provided both upfront capital and some minimal long-term operational funding. No long term sustainable source of funding could be assumed and thus the problem was one of balancing the capital acquisition costs of hardware (and in essence performance) against the facility implementation cost and the long-term operational costs including both power and personnel. Vendors were presented with the details of the physical space that had been acquired (empty slab space) and local power availability ($\sim$4MW) along with a fixed budget for five years of both operating and capital funding.

The SciNet RFP was comprised of the three main computing infrastructure components along with requirements for construction of the data centre. In order to maximize the flexibility and allow for innovation at the vendor level, the barest minimum specifications for the systems were given. Initially a two-phase deployment was planned, but discussions with vendors led to a single phase installation with a moderate budget withheld for future storage upgrades.

The minimum specifications for the GPC, which was to have the largest user base and would support the most varied science projects, were structured to be capable of simultaneously running three key types of applications: serial, MPI and large-shared memory jobs. Serial jobs were expected to use up to 75% of all the available cycles on the GPC. The ATLAS component (discussed further in Section 8) needed to be able to run up to 4,000 serial jobs on an x86 Scientific Linux 4.0 or equivalent platform. Parallel MPI jobs were expected to be able to use up to at least 10,000 cores for a single job and up to 25% of available cycles. Large, shared-memory, serial or OpenMP jobs were expected to use up to 16 cores and 128-256 GB of RAM.

No particular network infrastructure was specified and the possibility of an inhomogeneous cluster was allowed. A Linux-like environment, provisioning on demand and an integrated scheduling system were preferred. The tentative minimum specifications are outlined in Table 1

The TCS was envisaged to serve a more restricted community of users who required a system able to efficiently integrate models in which the coupling between processes was extremely tight. These included coupled climate system simulations as well as other problems in which hydrodynamic processes played a central role. The installed system was expected to be able to deliver at least 20 Tflops if the architecture was of a modern parallel-vector design in which multiple CPU nodes are interconnected at high-bandwidth or a higher peak processing capacity if some other architecture was proposed that could deliver an equivalent capability in terms of work load. Two benchmarks were provided for vendors that were expected to be performed on

their proposed solutions with the performance results playing a key role in selecting the winning bid.

Numerous criteria were specified for the storage subsystem but the overarching preferred guideline was to have a high-performance single subsystem accessible by all nodes in the GPC and TCS. Five main storage areas were envisioned: user `/home`, fast parallel `/scratch`, longer term storage on `/project`, a small space for backups and finally dCache space for the ATLAS project. The approximate delivered sizes for these areas in Phase I was expected to be at least 20, 600, 200, 100, 500 TB respectively (total of 1.5 PB) and in Phase II, 50 TB, 1.5 PB, 2 PB, 100 TB and 1.3 PB (total of 5 PB).

As a large amount of storage was being requested, allowance was made in the RFP for lower cost, high-density enterprise SATA-style drive technologies or equivalent to be used, but with a RAID6 or equivalent technology. Given the diverse populations to be served, specifying the exact performance criteria necessary was deemed too difficult and user case scenarios were presented instead give some approximation of the expected access patterns and performance required for various scientific projects. A rule of thumb was that all processing cores in the GPC should be able to simultaneously access the storage system with at least 1 MB/sec read/write.

The SciNet GPC was to provide the ATLAS Tier-2 Analysis Centre and, as such, had certain requirements that were specified in the RFP and that are summarized in Section 8.1. Vendors were asked to describe their technique and hardware and software requirements for implementation of dCache and how to integrate and interface this subsystem with the large SciNet disk storage and the component of the GPC capable of running the Atlas software.

The final component of the storage request was for tape backup solutions with a minimum of four LTO-4 or equivalent tape drives and 200 slots.

To ensure that the consortium received the best possible value for money it was left to vendors to complete this optimization by either working together in partnerships or building a consortium of their own to address the various diverse computational, data storage, networking and facilities requirements. This resulted in several unique and unusual pairings of vendors, but in the end IBM presented the most complete solution that achieved the goals of SciNet.

To complete the optimization of the solution IBM brought on board several partners which included:

- Ellis Don – General Contracting
- WZMH Architects – Architectural
- Hidi Rae Engineering – Mechanical Engineering
- Lapas Engineering – Electrical Engineering
- Modern Niagara – Mechanical Contractor
- Guild Electric – Electrical Contractor
- Johnson Controls – Chiller & Automation

This team completed several iterative design runs and cost projections for various datacentre configurations, efficiencies and long-term power costs to design the optimum physical facility, each time working along with the IBM HPC Engineering team to re-calculate the additional computational capacity that could be added after each round of optimization and cost savings, this in turn typically increased the electrical and mechanical load until finally a balance was reached.

It was clear that presenting the opportunity to optimize the complete facility and solution as a single business case allowed all vendors to provide the maximum possible return on our investment, with all vendors capable of negotiating with their individual partners in a more powerful way than SciNet could have one at a time. This forced close integration also resulted in much more tightly integrated solutions being presented to SciNet, with little wastage, bloat

Figure 2. Partial view of the computer room in the SciNet data centre. In front, the GPC iDataPlex racks; at the very back, the TCS and disk.

or unnecessary equipment or elements in any of the proposals. While each vendor approached the problem with a different strategy overall, several unique and compelling solutions were presented, but IBM was selected in part due to a slightly higher level of integration between the components, particularly at the support level.

Throughout the process all vendors were challenged to demonstrate that a very high degree of due diligence had been undertaken in forming their estimates, particularly around estimates of power consumption, efficiency and future power costs. Of these, only future power costs have been left to market forces, with IBM committing to both consumption and overall facility efficiency targets as part of the contractual obligations. The price of power, however, is beyond the scope of what most large IT vendors will accept in terms of operating risk and SciNet is responsible for this portion of funding.

4. Resulting System Architecture

The resulting system designed and delivered by IBM consisted of two main compute elements, a single shared storage element, all connected with a primary 10G Ethernet network, and sub GigE and Infiniband networks. The Tightly-coupled Capability System (TCS) consists of 104 IBM p575 nodes each with 16 Dual-Core Power6 CPUs at 4.7 GHz and 128 GB of RAM connected with DDR Infiniband, totaling 3,328 cores running AIX 5.3L as the operating system. The General Purpose Cluster (GPC) consists of 45 racks of 84 IBM iDataplex dx360 M2 nodes each with 2 Quad-Core Intel Xeon E5540 CPUs at 2.53 GHz and 16 GB RAM, totaling 3,780 nodes and 30,240 cores. The entire GPC is connected with Gigabit Ethernet, and 864 nodes (6,912 cores) are also connected via DDR Infiniband. The operating system is Linux CentOS 5.3.

The primary disk storage is provided by two DataDirect Networks (DDN) DCS9000 couplets with 1790 1 TB SATA hard drives, for a total usable space of ≈1.4 PB. The primary file system mounted by all compute nodes is IBM's General Parallel File System (GPFS, [32]) with three primary partitions; `/home` 14 TB, `/project` 365 TB, and `/scratch` 465 TB. The remainder of the disk space is used for the ATLAS project and under dCache. The systems as currently set up are shown in Figure 2.

The job scheduling software used is Adpative Computing's Moab Workload Manager [33] with the open-source TORQUE as its resource manager on the GPC and IBM's LoadLeveler on the

TCS. The Extreme Cluster Administration Tookit known as xCAT [34] is used for provisioning the operating systems of both the TCS and GPC nodes as well as the various infrastructure nodes.

5. Data Centre and Power Efficiency

The cost – and time required to build – power infrastructure downtown, combined with the lack of available quality space for a data centre (requiring lots of square footage, high ceilings and flexibility for mechanical systems) drove SciNet to actually place an advertisement in a national newspaper in its search for space. Any plausible rental cost, even going out 10 years in the future, would have been less than the one-off price of bringing a power feed of the size required into downtown Toronto. Thus, the data centre and the main SciNet offices are not co-located.

In early 2008, SciNet rented 12,000 square feet of space in an industrial/business warehouse complex roughly 30km north of the main University of Toronto campus (St. George campus). The current rental agreement is for five years with two options to renew for a further five years. The space is self-contained, has its own double-wide loading dock and two exterior walls with an under-utilized high-voltage power line running immediately along one of the exterior walls and a dedicated, separately metered 600V, 4,000 A power feed. The building construction is single storey, slab-on-grade with 15 feet from slab to beams with a metal slat roof above.

The rented space originally had no interior walls and the landlord was amenable to modifications of the exterior walls and roof which gave great flexibility in the the design of the datacentre and the location of the power and cooling infrastructure. The electrical (270 sq feet)and newly-constructed mechanical (1,900 sq feet) rooms are located along an exterior wall in such a way that the entire chiller assembly (shipped on a 40'x15' skid) had to be moved only a short disatnce directly from the loading bay into its final location. The cooling tower sits on a reinforced section of the roof immediately above the chiller. Two offices provide adequate room (360 sq feet)for staff who are usually on-site only to perform maintenance functions in this lights-out datacentre.

The computer room itself is a 3,100 square foot (290 m^2) rectangular room surrounded by corridors on all sides and has an 18" raised floor used only for the water piping, power and network cables. Transfomers (600V to 480V and 600V to 208V) are located on the exterior side of the machine room walls (in the hallways) with the corresponding distribution panels on the interior side of the wall. Three 35-ton air handlers provide air-cooling (on top of the raised floor) to the one row of racks which is not completely water-cooled (disk storage, switches and the Power 6 systems which are 25% air-cooled).

A significant research area being addressed with the SciNet machines is that of climate change and global warming, which is why creating one of the greenest datacentres in the world was of key importance in this project. This concern drove the design towards using water-cooling and a cooling tower. A traditional modern datacenter generally uses at least 33% of the energy going into its centre for cooling and other non-computing power consumption; however, SciNet and IBM have successfully created a centre that uses less than 20% towards these areas.

5.1. Water Cooling

One major design consideration for the facility was the appropriate heat extraction system. As the overall solution was approaching 2MW in power consumption the efficiency and floorspace dedicated to the cooling system was significant. IBM considered several cooling options, including traditional Computer Room Air Handlers (CRAH), compact In-Row Air Handlers, Above Rack Air Handlers, and a variety of coolants including phase-change materials and refrigerant based solutions. The use of water to the equipment was a direct result of the ease of implementation, relative safety of the fluid, and excellent heat transfer properties when correctly maintained.

Figure 3. The 735T centrifugal chiller used in the SciNet datacentre, with some of the associated plumbing.

IBM chose to use direct-to-rack water cooling for the vast majority of the equipment in the facility. The TCS was based on the IBM POWER6 p575 and required direct water cooling both to the rack via a rear-door heat-exchanger and to the processors themselves via a direct contact heat plate and appropriate tubing. For the GPC IBM proposed the use of the innovative iDataplex rack form-factor which had the capability to support a very large rear-door heat-exchanger capable of removing up to 30kW per rack when properly configured.

The decision to use water as the primary coolant was cemented by the relative ease by which SciNet was able to implement a "free-cooling" solution to further augment the overall efficiency of the facility as discussed later.

This decision was not without challenges however. The primary chiller for the facility is a 735T centrifugal chiller shown in Figure 3 with variable speed drive and an optimum output temperature between 4°C–7°C. To ensure the chiller could operate most efficiently the highest temperature that could be used would be 7C, which itself was too low to be used directly in a rear-door heat-exchanger (see Fig 4. The IBM rear-door heat-exchanger design is a remarkably simple one, using basic copper tubing and fins to create a very large, passive radiator with no moving parts, and no components other than the inlet and outlet Parker quick-connects. This passive design results in excellent reliability and very low cost, but also raises the potential for condensation should the inlet temperature drop below the facility's dew-point [35].

The same challenge is not present on the POWER6 hardware, which has the ability to self regulate with redundant built-in active heat-exchangers at the bottom of each POWER6 rack. As a result the two major computational elements of the SciNet complex are fed by two different water loops – the POWER6 equipment is directly fed the 7°C water from the chiller, and the iDataplex equipment is fed from a secondary loop moderated via a large heat-exchanger in the mechanical room to a stable 12-13°C. Further control of the dew-point in the room is completed by using active humidity management in one of the three 50T air handlers that provide air cooling to the small amount of equipment that was incompatible or impractical with rear-door heat-exchangers (45U non-standard storage rack, network switching with large amounts of cabling, etc.).

The result of the dual-loop system, with a very small amount of humidity control is that all heat from the 30,000+ Intel Cores in the GPC, and about 80% of the heat from the 3,300 POWER6 cores is directly transferred into the facility water supply and later returned to operating temperature via the cooling towers and 735T chiller. The flow-rate of the water through the system is carefully controlled to optimize the heat-removal against the operating

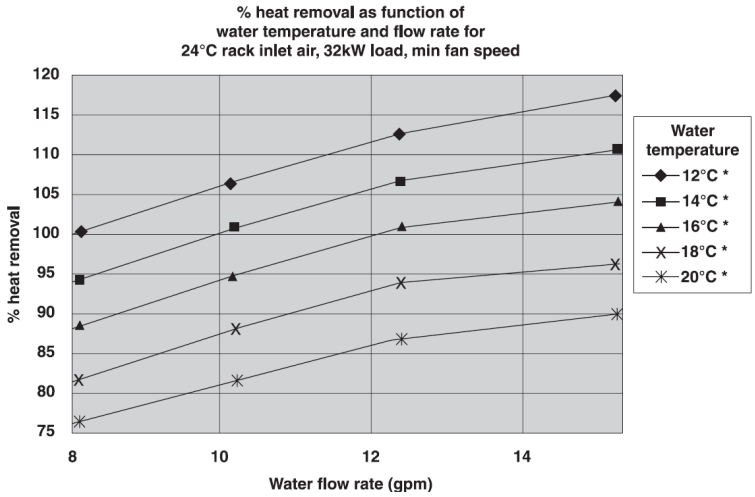

Figure 4. Typical performance of the heat exchanger, 32 kW heat load; from [35]

Figure 5. Cooling pipes being laid out below what will be the machine room floor during the construction of the data centre.

temperature of the equipment, with known capabilities for the rear-door heat-exchangers.

All cooling infrastructure was located below the floor, allowing plenty of space for the large schedule 40 piping required to handle the water flow as shown in Figure 5.

It is this aggressive use of water cooling that permits such a high efficiency for the overall data centre. Without the water cooling, the air temperature in the datacentre can reach 50°C in 15 minutes; thus any failure (or maintenance) of the cooling requires a rapid shutdown of the entire data centre.

5.2. Free Cooling

One of the major design elements of the solution was the overall solution efficiency. To attain a target PUE of 1.16 it is necessary to exercise all available natural benefits. The SciNet facility is located at approximately 43°50N latitude and lends itself to the potential for significant periods of time with temperatures capable of supporting a variety of "free" cooling methods. During the design of the facility the potential for augmented natural cooling or any form of economization was quickly identified with several options evaluated including:

- Air Economizers (Air exchange cooling)
- Water Side Economizers
- Heat Transfer to Local Business (operational cost offset)
- Phase-change Solutions (Stored Cooling)

To determine the viability of free cooling solutions the first analysis was to determine the climate patterns in the vicinity of the facility. While part of the SciNet research group includes significant expertise in climate science it was extremely easy to obtain local temperature profiles through general HVAC suppliers. Local temperature data is available at a fairly granular level in most of North America from ASHRAE certified HVAC suppliers as it is commonly used in the "Bin Method" for calculating local heating and cooling costs, where hours spent in temperature bins over the course of a year allow you to estimate the amount of HVAC required; tabular data for Toronto is given in Table A1 in the Appendix.

This data helps guide the possible use of ambient cooling, in this case the data shows that of the 8760 total yearly hours, on average 2811 hours are spent at or below 0°C/32°F and as many as 5529 hours are spent at or below the maximum water temperature of 12C/53°F. Put differently this indicates that it is reasonable to expect that approximately 32% of the year is spent below 0°C/32°F and upwards of 63% of the year is spent with external temperatures near our highest water feed temperature. The opportunity to benefit from the use of this ambient temperature profile is enormous.

As the decision to utilize water cooling to the racks had already been made, in part for density reasons, the use of air-side economizers and external air-exchange systems was deemed too costly, and complicated for this environment. For many environments this would make perfect sense and is actively being researched and developed by many vendors including IBM and Intel Corp. One particular area of continuing research is the long-term effect of air impurities, be they chemical or particulate in nature and whether this has any impact on the lifespan or reliability of the equipment. In essence a better understanding of the effect of these impurities would help guide the implementation cost of these solutions as extensive filtering would increase the cost. We did not pursue this option.

The extensive use of water suggested we primarily focus on water-side economizers, heat-transfer or sale to local business (as a cost mitigation strategy) or phase-change solutions. As the facility is in an industrialized neighbourhood a search was initiated to determine the viability of heat-sale. Unfortunately distance issues and the relatively low ΔT between inlet and outlet temperatures also eliminated this option. Evaluation of Phase-change solutions was also completed with IBM's expertise resulting from the implementation of a large phase-change solution at IBM's manufacturing facility in Bromont, Quebec [36]. However, while this design holds significant potential for the type of variable commercial workloads, the magnitude and consistency in the cooling requirements at a large HPC facility would have required significant phase-change infrastructure and a longer term ROI horizon than the 3-5 years necessary for this type of facility.

Upon evaluation of all other options it was clear that water-side economization was the most appropriate path forward. Put simply, a water-side economizer uses just the pre-existing pump hardware of the mechanical chiller solution to ensure proper fluid flow, but bypasses the chiller

itself, relying instead on just the external indirect cooling tower for heat removal. This bypass can be configured in a variety of modes, variably adjusted to provide partial, or full free-cooling depending on the external ambient temperature. While it is possible to begin to leverage this ambient cooling any time the external temperature is at or below the 7°C temperature of our primary cooling loop the mechanical bypass systems inherently increase the complexity of the solution and introduce additional potential failure modes. The potential for issues with the transition from total mechanical (chiller) cooling to partial and then full free-cooling increase proportionally to the number of transition events. As a result in practical application use of the water-side economizer bypass tends to be limited to sections of the year where average temperatures approach 3°C or below on a routine basis, and after transition to full free-cooling minimal cycling back to mechanical cooling is experienced. This decision has worked well, providing an excellent balance between efficiency and the risk of failure associated with the many automated mechanical valve changes that must occur to completely bypass the main chiller.

5.3. Power Metering

In order to assess and monitor cooling efficiency in a datacentre it is critical to meter and record as much data as possible. A total of 22 circuits are metered in the SciNet datacentre in order to separately monitor the power used by the water pumps, chiller, cooling tower, lights, UPS, the TCS (in groups of 2 racks) and the GPC (in groups of 3 and 4 racks). This granularity is fine enough to measure the PUE.

A sample weekly plot is shown in Figure 6; during this period one can see that the GPC typically draws ∼750kW, the TCS draws 400kW and the PUe varies between 1.12 (corresponding to a roughly 6 hr period overnight Thurs-Fri during which the outside wet-bulb temperature was low enough for the system to use free instead of mechanical cooling) and 1.25. Data over an entire year of operation is currently being analyzed and appears consistent with an average PUE less than 1.2.

The SciNet datacentre qualifies for hourly spot-market electrical rates as a commercial customer using more then 50kW but less than 4MW of power. Rates are set by the Independent Electrical System Operator (www.ieso.ca) and the average weighted price YTD (Oct 2010) is 4.0 cents/kw-hr. Unfortunately, this basic consumption charge averages out to just 45% of the actual power bill which includes various charges for debt retirement, transmission and, most significantly, to account for the difference between the spot market price and the rates paid to regulated and contracted generators. Total datacentre power consumption for the first full year of operations (1 Aug 2009 to 1 Aug 2010) was 11.8M kW-hrs.

6. Network Overview and Design

The SciNet computing facility is comprised of multiple networks that are integral to the overall functions of the cluster, storage and data centre itself.

The management functions of the clusters and storage as well as infrastructure control and monitoring elements of the data center are handled over multiple subnets. These are mostly Gigabit Ethernet networks that run over Blade Networking Technologies (BNT) G8000 class switches and converge in a Force10 C300 Director class switch.

A high-performance low-latency network connects nodes of the TCS POWER6 based p575 cluster together. The network is a quad-plane DDR Infiniband network utilizing QLogic 9240 Director class switches. The network is configured for full bisectional bandwidth and currently provides an 80 Gbps connection to each node but has the capability of being upgraded to 160 Gbps per node. This network serves as both the message passing interface for parallel codes as well as providing data access to the GPFS filesystem for the TCS Nodes.

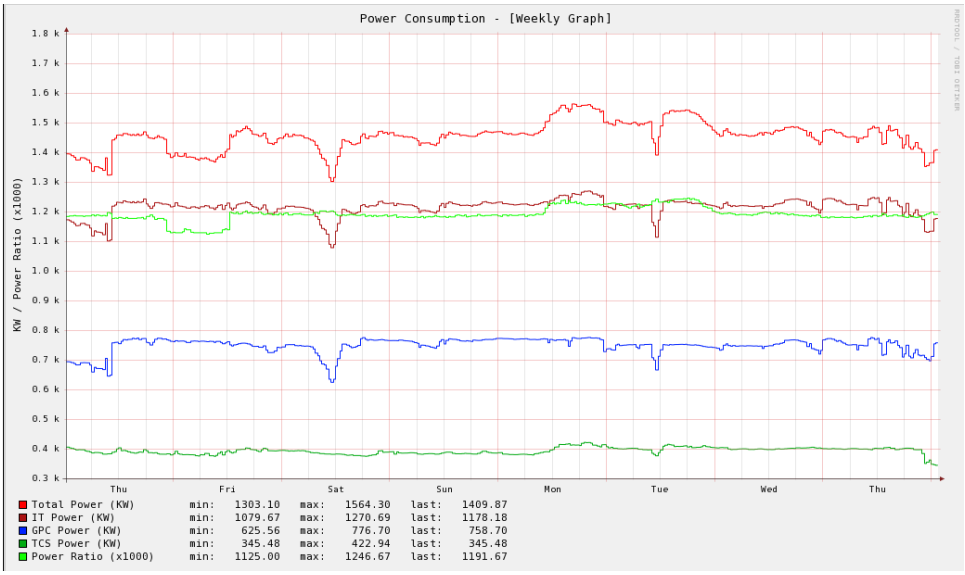

Figure 6. Power consumption (in kW) over a one week period by, from top to bottom; entire datacentre, IT equipment, GPC and TCS. The light green line is the PUE (ratio of total power to IT power) which varies from 1.125 to 1.25 over this period

Analysis of all the potential user codes that would run on the GPC at SciNet showed that a large portion of the expected workload would be actually be suitable to a lower bandwidth, higher latency network. However, there was still a requirement for some codes to have access to a high-bandwidth low-latency network. Based on this analysis it was decided that the optimal network design for the GPC would actually be a hybrid of two networks. As costs of high-bandwidth low-latency networks can approach up to 30% of the overall cost of a cluster, this tailoring of the networks to meet the workload requirements allowed SciNet to procure a much larger cluster than if it was decided to build a uniform high-bandwidth network to all the nodes. The network design for the GPC connected all of the nodes to a Gigabit Ethernet Network and almost 25% of the nodes to a full-bisection DDR Infiniband network configured in a CLOS Fat-Tree topology. This 2-tier DDR Infiniband network utilized Mellanox Connect-X Host Channel Adapters (HCA) connected to QLogic 9024 Leaf Switches which then uplinked to QLogic 9240 Director Class Core Switches.

By far the most interesting network in the SciNet facility is the Gigabit Ethernet network that connects almost four-thousand devices, including all of the compute nodes in both the GPC and TCS clusters. While each node is only connected by a Gigabit connection, the sheer number of devices leads to challenges in designing a network that can sustain a reasonable throughput for all of these nodes concurrently. Aggregation can be accomplished by utilizing 10-Gigabit Ethernet between switch layers but very few director class Ethernet switches can handle full bandwidth at this scale. At the time of the acquisition of the GPC, the Myricom Myri-10G line of switches were the only 10-Gigabit Ethernet switches on the market to support full-bisectional bandwidth across all of their ports. However, this extreme level of performance carried with it some trade-offs. The most important of which was that they only supported flat layer 2 networks. No higher level OSI function such as routing or even VLAN support was present. This limitation is what fundamentally led to the SciNet GPC compute network being constructed as potentially the largest flat layer 2 network in the world with approximately eight-thousand MACs (four-thousand nodes, each with a host and IMM MAC). The challenges

and lessons learned from implementing, maintaining and supporting this network have provided deep insight into scalability of the Ethernet protocol at its limits.

The GPC Gigabit Ethernet compute network is configured as a 2-tier network with forty-two nodes connected by Gigabit Ethernet to a BNT G8000 Series switch. The G8000 is then uplinked to a core Myri-10G switch over a single fibre optic 10-Gigabit link. This is configuration is repeated ninety-two times for a total of 3,864 GPC compute nodes. Additional infrastructure nodes and all 104 TCS nodes are also connected to the GPC Network in this same fashion. The TCS nodes were added to this network in order to extend the same GPFS filesystems across both the GPC and TCS clusters. A common network is required for this in order to support token passing, locking and other miscellaneous functions. The GPC network is not a full-bisection network - blocking of approximately 4:1 is encountered at the BNT G8000 edge switches as the Gigabit links are aggregated into a single 10-Gigabit Uplink. Even with this moderate level of blocking the GPC compute network provides a massive amount of bisectional bandwidth for an Ethernet network. The design of this network and it's high throughput and modest blocking factor was a significant contributing factor in the achievement of this cluster's 16th place ranking in the June 2009 Top500 list and it's distinction as the only Gigabit Ethernet cluster in the top 80 list entries.

6.1. Static ARP Tables

One of the considerations in building a large flat Ethernet network is handling broadcast traffic. On smaller networks this is usually not a concern as broadcasts are typically a relatively small percentage of the overall traffic. However, as the number of endpoints in a network grows, the amount of time a Network Interface Controller (NIC) spends processing broadcast traffic from all the other endpoints grows linearly. As any given NIC has a fixed amount of bandwidth and processing capability the amount of time spent processing broadcasts grows as an N problem. To compound this issue, the internal architecture of the Myri-10G switch which provides the high-bandwidth full-bisection Ethernet network, does not perform as well with excessive broadcast traffic.

On the SciNet GPC network the most significant contributor to broadcast traffic was the Address Resolution Protocol (ARP). This protocol resolves IP addresses to MAC addresses for unicast communication in a local subnet. It was found that as the the number of endpoints on the network crested beyond 1500 nodes or 3000 MACs, ARP traffic started to consume a significant amount of each NIC's processing resources and led to issues such as flow-control lock-ups and the NC-SI hangs that are described later in this document. In order to reduce the overall amount of broadcast traffic on the network, it was decided that we would implement and maintain a Static ARP table that contained mappings for every node in the network. While this is not practical for many networks, it is perfectly feasible for a tightly controlled internal network where endpoint devices and not leaving and joining on a constant basis. With the Static ARP table inserted into the ARP cache of every node as it booted the broadcast traffic on the network due to the ARP protocol was essentially reduced to zero. In order to handle the general housekeeping of node and NIC hardware failures and replacements, a few small program were written in order to maintain the static ARP table and ARP caches across all the nodes.

- `./replacenode` - A script written to clean out the xCAT database and Static ARP table of a particular node that was being removed from the cluster. This is mostly used for hardware service replacements that would effect the MAC of the node (i.e. System board and/or NIC).
- `./updatearp` A script that updates the static ARP table and dynamically updates the ARP Caches of all running nodes with the ARP entry of a newly discovered node. This is used after HW replacement of a system board or NIC.

The implementation of the Static ARP table was probably the single most important configuration change in stabilizing the GPC network for production use. It is possible that in the future, with some of the planned upgrades, the static ARP system may no longer be a requirement. However, currently it is still in full production use.

6.2. Other Network Considerations, STP and Flow-Control

Another lesson were learned during the deployment of the GPC network was that a slight misconfiguration in one of the standard Ethernet protocols, that may not affect a smaller network, could have a dramatic effect at this very large scale.

By default all switches in the SciNet GPC network were enabled with Spanning Tree Protocol (STP). This protocol searches for and eliminates 'loops' in the network. We found that even with RSTP (rapid spanning tree protocol) the amount of time it took the network to converge could cause interruptions to jobs and filesystem I/O in the network. This problem was exasperated by a misconfiguration on a few ports that had nodes attached but were configured as switch to switch links (i.e. not 'edge' ports) This had the effect that anytime this link was reset for any reason it forced the entire network to run STP and re-converge. One of these ports happened to be connected to a p520 GPFS NSD Server which was having issues with momentary link resets (described in section 7.3). Since this port was not configured as an edge port, the network would end up in regular STP loops trying to re-converge before the next link reset. Once these configuration issues were found and corrected the instances of STP having to re-converge the network were significantly reduced. However, in the end it was decided to disable STP completely across the GPC network. The reason for this decision was that the stalls in network traffic caused by STP outweighed its potential benefit of finding and correcting a loop in the network. As the physical topology of the GPC network was well understood and is of a relatively 'simple' repeating design, the likelyhood of introducing a network loop was extremely low. Local security at the facility also makes it unlikely that anyone will connect a rogue switch into the network and create a loop.

Another challenge encountered scaling up the GPC network for production use was the implementation and control of the 802.3x flow-control protocol. Flow-control is an extremely valuable low-level protocol that allows network devices to limit in-bound traffic rates by issuing pause frames to stop from becoming overloaded. This is much less costly than a network device dropping packets and relying on higher levels of the network stack, like TCP/IP, to request re-transmission of the lost data. The potential pitfalls of flow-control is that rogue or misbehaving devices have the potential to lockup the entire network by continuously sending pause frames. On small networks this is unlikely. However, at the scale of the GPC network, with its massive amounts of aggregate throughput and sheer number of endpoints it becomes statistically more likely that a network device can become overloaded or even enter an unstable state. This was seen on multiple occasions during the implementation of the GPC network. The first observed occurrence of this was when a compute node encountered a rare hang condition in the Network Controller Sideband Interface (NC-SI) between its host NIC and IMM baseboard management processor (see Figure 7. When this NC-SI channel locked up and the host NIC could no longer pass packets up to the IMM processor, the buffers on the host NIC filled and then it started flooding the network with pause frames essentially locking up the entire network.

After an in-depth investigation it was found that excessive broadcast traffic was leading to the hang condition of the NC-SI channel. This discovery was one of the original motivators for the development and implementation of the Static ARP table in the GPC cluster. IBM later released updated uEFI and IMM firmware revisions that stabilized the NC-SI channel and made the likely hood of these types of lockups much less likely.

As the potential for excessive flow-control pause frames from rouge or misbehaving devices always exists on the GPC Network additional steps were taken to ensure that a single device

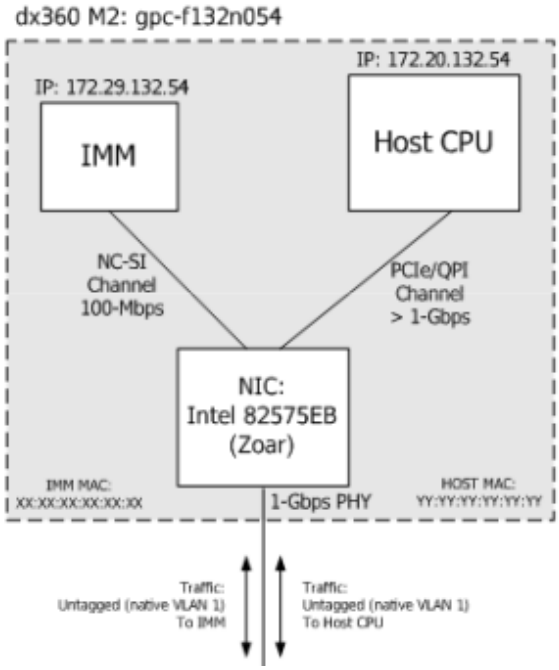

Figure 7. Schematic showing the Network Controller Sideband Interface (NC-SI) between host NIC and IMM baseboard management processor.

could never lockup the entire cluster indefinitely. In working with the development teams from Myricom an additional function was added to the Myri-10G switch which allows it to monitor the percentage of pause frames that it receives on any given port and disable flow-control on that port if it exceeds a configurable limit. With flow-control disabled the port would start dropping packets as opposed to forwarding pause frames. This would allow the rest of the network return to normal operation while only devices on the given port would be effectively blocked out until the flow-control situation was resolved, at which point the switch would re-enable flow-control on the port and return it to the network. This configuration parameter, now available on all Myri-10G switches, is:

```
PausedPercentLimit = 63
```

where the setting is in one-hundredths of a second and represented in hexadecimal; 0x63 is 0.99 or roughly 1 second.

6.3. 802.3ad, Myricom, and System P

In order to increase bandwidth through to the GPFS filesystem from the GPC network, multiple 10-Gigabit Ethernet links from each GPFS NSD Server were 'bonded' together using the 802.3ad Link Aggregation Control Protocol (LACP) and connected directly to the Myri-10G switch, as opposed to going through a BNT distribution layer. Early performance testing showed that downstream traffic from the GPFS Servers to the compute nodes in the GPC network scaled with the multiple links. However, the opposite was not true. Upstream traffic from the compute nodes to the GPFS Servers never exceeded the capabilities of a single 10-Gigabit link. After much investigation it was discovered that the MAC addresses of all of the compute nodes were even in the 6th (last) hexadecimal octet. This was the result of the MAC burn-in process used by Intel for their dual-port 82575EB controller which was the on-board NIC of every compute node.

The first five octets of both ports were identical and the last octet always varied by one with the first NIC port always being even and the second NIC port always being odd. Since all compute nodes were connected to the first NIC port this meant that all the MAC addresses were even. The Myricom protocol for LACP balancing did a simple modulo 2 on the source MAC, and since the result was always 0, the traffic was always balanced to a single 10-Gigabit port on the GPFS Servers. Once this was identified, Myricom modified their LACP balancing algorithm to hash the entire MAC address before determining the modulo 2 result. This resulted in good balance of upstream traffic and increased bandwidth to the GPFS Servers. This firmware updated again made it into the production stream and now is part of every Myri-10G switch sold.

By far the most unique issue identified during the deployment of the GPC network was a small but troublesome link reset that occurred between the 10-Gigabit Ethernet NICs on the IBM System p520 GPFS NSD Servers and the Myricom Myri-10G switch. At first, with the STP Edge port misconfiguration, these link resets would cause STP to constantly re-converge the network. However, once the STP configuration was corrected and then ultimately disabled. The link resets became an oddity that didn't seem to have a major impact on the overall network deployment. It wasn't until the GPFS IO testing ramped up that it was determined that the link resets were having a negative effect on the overall performance of the filesystem. Engineers from both IBM and Myricom engaged to identify and resolve the issue. Testing was done with the p520 10-Gigabit adapters and other vendor's switches as well as other 10-Gigabit Ethernet adapters and the Myricom switch. Issues were only ever seen with the combination of the p520 10-Gigabit adapter and the Myri-10G switch. After using an inline network sniffer/analyzer on a similar setup at the Myricom labs it was found that the p520 10-Gigabit adapter was releasing a packet that should have been filtered out and that the Myri-10G switch was miss-interpreting this packet as a link error and cycling the port. Firmware was developed and provided for both the Myri-10G switch to correct the packet miss-identification as well as for the p520 10-Gigabit adapter to filter out the packet before it ever hit the wire. Applying either of these updates would resolve the link resets but both were applied to the SciNet system and both fixes are now part of IBM's and Myricom's standard firmware releases.

6.4. Future of the GPC Network: Projects and Investigations

Two projects are currently under investigation for the future of the GPC Ethernet Network. The first project is looking at doubling the number of 10-Gigabit uplinks from the ninety-two BNT G8000 switches. This would effectively half the overall blocking factor of the network and bring it close to 2:1. Enough ports are available on the core Myri-10G switch to support this upgrade and the BNT can support up to four 10-Gigabit uplinks. It is planned that smaller scale testing will be completed on a portion of the GPC cluster to determine what effects this increased bandwidth will have on a variety of workloads. As the BNT G8000 can support the additional 10-Gigabit uplinks in a variety of configurations it is expected that several of these will be explored during the small scale tests:

Two 10-Gigabit uplinks from the same dual-port XGE uplink module, which connects to a single Broadcom ASIC inside the switch. Uplinks will be configured in an 802.3ad LACP bond to the Myri-10G

Two 10-Gigabit uplinks, each from distinct dual-port XGE uplink modules, which connect to separate Broadcom ASICs inside the switch. Uplinks will be configured in an 802.3ad LACP bond to the Myri-10G.

Two 10-Gigabit uplinks, each from distinct dual-port XGE uplink modules, which connect to separate Broadcom ASICs inside the switch. The G8000 will be configured with an internal VLAN, so that 1-Gigabit ports on each Broadcom ASIC will route to the respective uplink module on that ASIC. This configuration essentially carves the switch into two smaller switches, each with their own dedicated ASIC and uplink module.

The second GPC Network project currently being investigated at SciNet is the potential reconfiguration of the IMM baseboard management controllers in the compute nodes to support VLAN tagging. Currently the GPC Network supports close to 8000 MACs. Approximately half of those MACs are the IMM modules in the compute nodes. The addition of a VLAN tag to traffic originating, or destined to, the IMM would provide the opportunity to segregate and divert this traffic at the level of the BNT G8000 switches before it reaches the core Myri-10G switch. This would reduce the number of MACs that the Myricom switch would have to support from 8000 to 4000. A schematic of the proposed change is shown in Figure 8.

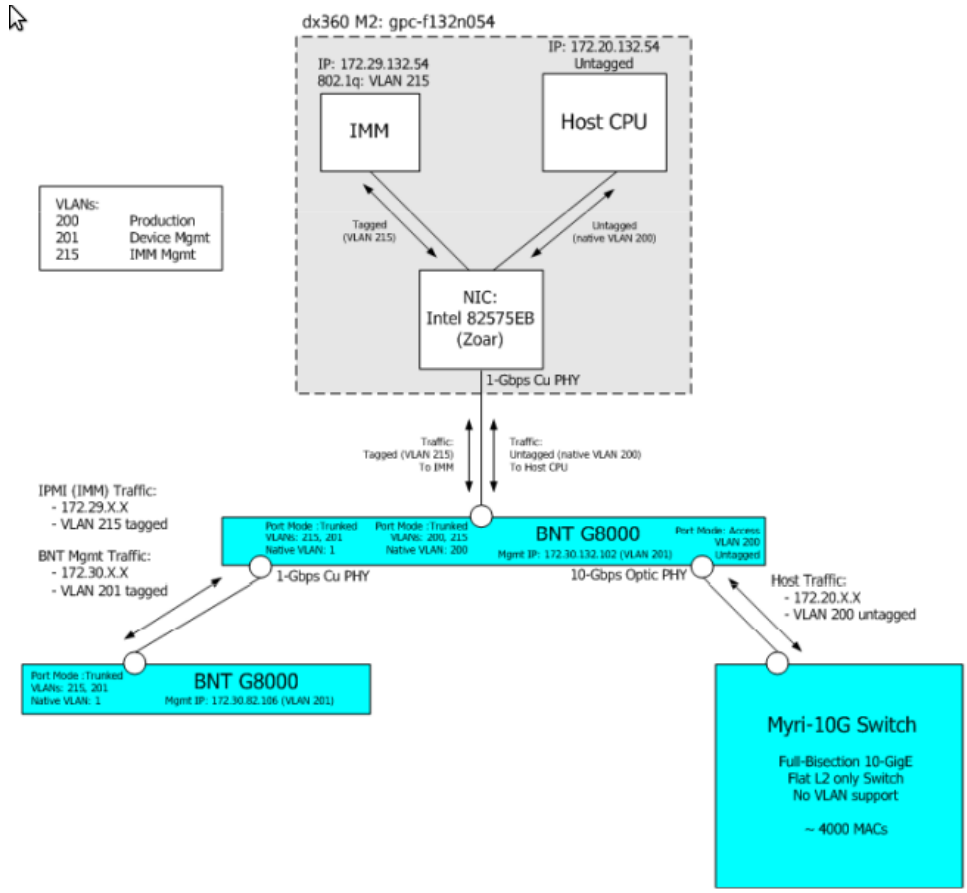

Figure 8. Schematic showing proposed VLAN tagging to segregate and divert IMM traffic at the level of the BNT G8000 switches before it reaches the core Myri-10G switch.

This reconfiguration would require minimal additional hardware to support the new IMM VLANs and could have an impact on further reducing occurrences of broadcast storms and excessive flow-control on the Myri-10G. This would ultimately benefit both the end user and filesystem traffic for the GPC Ethernet Network.

There are currently some minor technical challenges with the Linux device drivers for both Intel and Broadcom NICs in this environment. Default drivers bundled with current Enterprise Linux Distributions filter VLAN tags once they are loaded. This effectively cuts off the out-of-band communication with the IMM once the Operating System is loaded. In lab testing it has been shown that the current versions of the Broadcom drivers do not exhibit this same behaviour, and the Intel driver can be set to bypass its VLAN tag filter in one of three ways. These OS considerations are not expected to cause any setbacks should SciNet decide to move

forward with this project.

At this point in the project, plans have been constructed to deploy this new architecture to the SciNet GPC Network and integrate it with the current xCAT management and infrastructure environment, as shown in Figure 9.

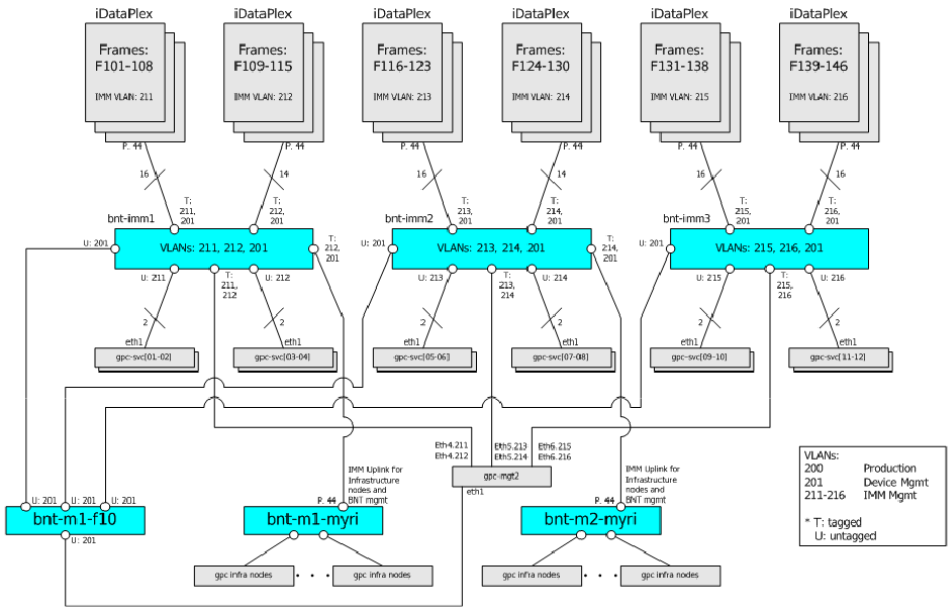

Figure 9. Schematic showing proposed VLAN tagging to segregate and divert IMM traffic at the level of the BNT G8000 switches before it reaches the core Myri-10G switch.

7. File System and Storage Design

7.1. Considerations for a Stateless, Diskless environment

The enormous scale of the computing resources at the SciNet facility demanded that alternative methods be considered for the deployment and management of software in the clusters. With just under four-thousand independent computing devices in the facility, a traditional deployment methodology of installing an operating system software stack to a hard drive on each individual node would present several administrative challenges. One of the most common and problematic issues for clusters of this size is software drift. This is the variability, over time, of the software and configuration files contained within each node. This most often occurs as a result of "silent" failures in bulk push updates of software patches and/or configuration files which are not accepted by just a few nodes out of the entire cluster. As these kind of updates are repeated over time the software variability across the cluster increases. This can eventually lead to symptoms such as job failures or "jitter" in the cluster.

Stateless computing resolves many of the administrative challenges of extremely large scale clusters by removing the stateful implementation of the operating system software from the hard drive on each compute node. Instead a common operating system image is developed on a management node and is sent out to many nodes as they boot from the network. The image is expanded and run from RAM in the node and the "personality" of each node can be determined by hardware unique identifiers such as the MAC address. Stateless computing can resolve challenges such as software drift and can also be a valuable tool to test and deploy software patches or even different customized operating system images. It is important to note

that stateless computing does not automatically imply that nodes are diskless. Hard drives can still be installed in nodes and used for other purposes, such as swap or scratch space. However, in extremely large scale deployments such as the GPC system at SciNet, Mean Time Between Failure (MTBF) and overall power consumption of the system can become a concern. One of the devices most prone to failure in a compute node is the hard disk, and the more drives you have the more likely you are to have a failure. As you approach four-thousand hard drives in a clustered system the MTBF can be measured in hours or days. In addition, four-thousand hard drives consume a considerable amount of power, their elimination from the cluster nodes contributed considerably to reducing the overall power draw of the cluster.

Implementing the stateless computing model and eliminating the local hard drives from the cluster compute nodes has not been without its challenges. The first challenge encountered was to deal with software components that require nodes to maintain state across reboots. One such package is GPFS, which requires that each node maintain an up to date copy of the GPFS cluster configuration files. It is not possible to include these files into the stateless operating system image as they are always changing. Instead a server was configured in the environment to provide the up to date GPFS configuration files to any node that requested them in real time. The stateless compute nodes were then configured to query this server for these files on boot before starting the GPFS daemons. This allowed the stateless node to have an up to date copy of the GPFS configuration files and successfully join the GPFS cluster on boot. It should be noted that other technologies are now available to handle stateful software requirements in a stateless environment. The xCAT Management tool that provides the stateless framework at SciNet now supports a new paradigm called "Statelite". This operating model combines all the benefits of stateless while allowing administrators to configure specific stateful files in an operating system image. Each node boots from the common stateless image but refers to its own copy of the specified stateful files. This implementation is being investigated by SciNet but is not currently in use.

Another challenge specifically related to the absence of hard drives in the GPC nodes is the handling of memory allocations without the benefit of a swap space. While it is fairly obvious that running high performance computing codes outside of the bounds of a node's physical memory capabilities is not very "high performance", not having the benefit of this buffer space changes a code from running slowly to outright failing. Interesting effects were seen on the GPC as user codes attempted to malloc() more memory than was physically available on the node. The Linux Out-of-Memory Killer (OOM) would hunt down processes to terminate in order to free memory for the requesting user process. Ironically, the process typically targeted was the GPFS daemon, which consumes moderate space as it allocates the GPFS pagepool (i.e. Filesystem cache) space in with the daemon. Killing GPFS would potentially allow the user application to complete its malloc() but it would be left without a filesystem to work with and would typically fail because of this. This issue has been somewhat mitigated by adjusting the OOM killer configuration to not target the GPFS daemon process. However, this is not a perfect solution as there will always be user codes that attempt to allocate more memory than is physically available on the system.

7.2. GPFS and "Green" Provisioning

The stateless capabilities of the xCAT Management tool when combined with Adaptive Computing's Moab Intelligent Cluster Scheduler allow the SciNet GPC to operate in a dynamic provisioning "green" mode. In this mode of operation, the Moab scheduler is able to instruct xCAT to provision certain nodes with specific operating system images based on either policies or criteria within the job. If nodes complete a job and remain idle with no new workload to backfill them, then Moab can instruct xCAT to power-off these nodes in order to reduce the overall power consumption of the cluster. If new workload arrives in the Moab queue and all

powered on nodes are busy then it can instruct xCAT to turn-on nodes that are powered off and provision them with the correct Operating System for the workload in the queue.

Several challenges were encountered in operating the dynamically provisioned environment at the large scale of the GPC System. Some of these issues were related to the Ethernet network and others were related to how certain software reacted to the dynamic environment at this scale.

Both computing clusters in the SciNet facility have common access through an extremely large Ethernet network. With approximately eight-thousand MAC addresses it is potentially one of the largest flat layer 2 Ethernet networks in the world. There were many considerations and technical challenges in building this network that are discussed in Section 6. As it related to dynamically provisioning the cluster, uncertainties in some network operations such as DHCP, TFTP, etc, could potentially lead to nodes not completing their boot and/or provision after being instructed to do so by Moab through xCAT. The Moab Service Manager (MSM), which is the interface between Moab and xCAT, uses a polling mechanism to query the state of nodes through xCAT. If a node did not completely boot then Moab would wait for the polling interval for provisioning to complete and then when it detected that not all nodes completed the provisioning it would start another cycle. This could lead to extremely long lead times before jobs began actual execution on the cluster and even could cause thrashing due to the occassional node boot failures when sufficiently large numbers of nodes were being provisioned.

Moab's MSM also instructed xCAT to probe all the Integrated Management Modules (IMM) on the compute nodes every 5 minutes. It used this information for multiple purposes including the "amber light avoidance" feature which alerted Moab to potential hardware issues on specific nodes so that they could be avoided for job scheduling. Due to the nature of the IMM and the network this IMM probing would lead to broadcast storms that would potentially disrupt all network traffic. The challenge of broadcast storms on the network has been one of intense focus throughout the bring-up and ongoing operation of the cluster. While attempting to avoid or minimize broadcast storms, the MSM polling intervals were gradually increased from 5 minutes to 15 minutes, and then 30 minutes. Finally MSM polling was disabled when the "green" mode was turned off. This has resulted in a much more stable network.

The final challenge with the dynamic provisioning environment was encountered in the shutdown procedure for the nodes. Once idle nodes had been inactive for a certain time period Moab, through MSM, would instruct xCAT to turn off these nodes through the "rpower" command which would be synonymous to pulling the power cord on a PC. Since the software stacks on these nodes were stateless it was originally thought that this would have no adverse effects. What was found through observation though was that the GPFS filesystem would need to run through a recovery process if a node in the cluster suddenly disappeared without properly exiting the cluster. This pseudo log rollback is done to ensure consistency of the filesystem. If many nodes were powered off within a close time span then this could potentially lock-up the GPFS cluster for other users while the recovery operations took place. This could cause jobs on other nodes to fail as they exceeded their own timeouts for I/O completion. While it was never fully tested, since "green" mode was disabled, modifying the Moab/MSM/xCAT interface to instruct a node to properly exit the GPFS cluster before executing the rpower command should provide significantly more stable operation.

As described above, there have been many challenges with the dynamic provisioning at this ultra scale. The increase in utilization of the SciNet computing systems has been so rapid that there has actually been very little opportunity for power saving gains from "green" mode operation, as almost all nodes are busy all the time. That being said, several projects are currently under investigation that should further improve the performance of the Ethernet Network, and this in conjunction with further optimizations of the Moab/MSM/xCAT interface should allow the "green" mode to be re-enabled in the future.

7.3. Managing GPFS

GPFS is a high performance clustered file system, and its managers (both for token and for file system traffic) play a critical role in the overall performance and stability of the SciNet cluster. Upon installation of the TCS (104 compute nodes), SciNet's eight NSD servers were used as the GPFS managers. This setup worked adequately without issues. However, file system hangs and inaccessibility started to occur after the addition of the GPC, which resulted in a 40-fold increase in the size of the GPFS cluster. It was only then that the GPFS support team identified that NSD servers cannot simultaneously function as filesystem managers for a large cluster. The first attempt to remedy this situation was to use KVM virtual machines on x3550's as the GPFS managers, which do not have very stringent hardware requirements. This approach, however, failed completely. In the end, the GPFS managers were moved to dedicated machines, which greatly improved performance. SciNet is currently using eight GPFS managers attached to two management switches.

There are many factors that affect filesystem performance. One of the most important is network congestion. GPFS relies entirely on the underlying network for all I/O and token communication. As described above, the network of the SciNet machines is implemented as one big flat network, a common practice for a GPFS cluster. However, due to the size of the cluster, which was at the time the largest single-domain GPFS cluster ever built, this design posed a huge challenge to SciNet. Network congestion, especially broadcast storms, brought down our GPFS cluster many times in the early days. IBM Canada's HPC team identified this issue and implemented static ARP tables in our system. This approach drastically reduced the network congestion, and GPFS stability was greatly improved.

In an effort to further reduce network traffic, SciNet reduced the use of Moab Service Manager (MSM) queries, which acquire information such as status and power utilization from each GPC node. This traffic goes through the management network, which shares the same physical wire as GPFS, and can therefore inadvertently affect the data network greatly. As mentioned above, disabling "green" mode helped reduce the load on the Ethernet network, thus increasing the stability of GPFS. As proposed by IBM Canada's HPC team, SciNet is currently investigating the possibility of improving the network, and hence GPFS performance, by separating management traffic from the data network through virtual LAN tagging.

Other factors that affect GPFS performance encountered at SciNet were:

- Time synchronization across the entire cluster.
- Critical GPFS parameter settings such as page-pool size and file system block size.
- Making sure that filesystems span a sufficiently large number of disk spindles in order to maximize both bandwidth and IOPs.
- Ensuring that all network equipment, such as NICs and switches, have up to date firmware that results in more stable operation.
- Aside from system manageability factors, GPFS is not immune to careless or misinformed users, who do not use the system efficiently. Educating users about the shared nature of the SciNet resources, particularly the filesystems, can weed out inefficient I/O practices, such as incurring in huge numbers of IOPs, reading and writing files too frequently and in small blocks, and creating millions of small files. At SciNet we are constantly monitoring the systems' CPU, memory, and I/O usage. We thereby identify issues and problem codes, and help users optimize their codes and I/O access patterns to improve their jobs' as well as the overall system's performance.

7.4. GPFS, Small File I/O, and Users

GPFS is designed and tuned to be a high-performance parallel HPC filesystem; that is, it makes possible very high-bandwidth parallel access to disk. As with other parallel HPC file systems,

it is designed for large block file I/O, and is not especially suited to sustaining large numbers of I/O Operations (IOP) that perform small reads and writes. This fairly straightforward and relatively unavoidable fact of computer systems engineering has caused the largest number of individual issues; not so much technical, but rather of an educational nature.

Some users, particularly those new to HPC, saw our systems as fairly familiar linux systems and thus believed they could do what they had been doing on their desktop — which, after all, worked[2] — and simply multiply that by 1000 jobs. We have seen users use P^2 files for communications between processes (writing "messages" from process i to process j in file ij, and using file permissions as a sort of semaphore system); we have seen users reading and writing individual floating point numbers to files from deep within their computational kernels. And of course, we have seen more mundane cases where each task writes a few hundred small (few kB) files, and then by multiplying this by 1000 or so jobs results in million-file directories or directory trees. In addition, users frequently write out real data as ASCII text, which is both typically written out in many small pieces, requires more disk space, and is significantly slower due to the cost of conversion to string.

Of course, if these practices only hurt the user's own performance it would not be a huge issue; however, unlike per-node memory or CPU usage, the filesystem is a shared resource, and the performance of the overall filesystem is extremely vulnerable to even a single user doing a lot of IOPs on even just a handful of nodes. Making this even more urgent in the first year of operation was that the large amount of excess network traffic from the few sources mentioned previously in this document, combined with heavy IOPs usage (on the GPC, all administrative network traffic goes over the same network as I/O – which, on the Ethernet GPC nodes, is also the MPI network) resulted in a filesystem which was far less stable than we would have liked with the older GPFS 3.2 series. The newer 3.3 series is significantly more stable, but the ability for a single user to starve out others for IOPs even from just a few nodes remains.

On a more mundane level, since everything on the two clusters shares a very small number of file systems, a few users with very many small files (and until the most recent series of user outreach campaigns, users with tens or hundreds of millions of files were not unusual) can enormously slow down important system processes which must traverse the file system file by file, such as backup, quota enforcement, and scratch purging. To help address that issue we recenty introduced strict quotas per user and per group on the amount of data and the number of inodes that are allowed on /scratch.

The above considerations have made very active and detailed system monitoring, so that particular users can be selected for focused and mandatory educational opportunities, crucial to ongoing operations. In some cases where little memory is used, user codes can be trivially modified to use ramdisk rather than local disk for high-IOP stages of computations, with final results being tar'ed and sent to GPFS in one operation. In other cases, code has to be significantly rewritten to perform well on shared filesystems (but this almost invariably results in improved operation even on desktop machines). However, one of our biggest users—the ATLAS collaboration—has code and workflow defined elsewhere and can not easily make such changes; for them, finding solutions was much more difficult.

7.5. Hierarchical Storage Management

Implementing a hierarchical storage management scheme (HSM) is a pilot project started in July/2010 with a select group of users, and is still in progress.

What we would like to offer users is a way to offload/archive data from the most active file systems (scratch and project) without necessarily having to deal directly with the tape library or "tape commands".

[2] or so they believed

In the initial stages we explored how far we could go by just extending the functionality of our current backup system, a Tivoli Storage Manager installation composed of a TS3310 tape library with 396 slots and 4x LTO4 drives.

IBM manuals describe a fully integrated TSM/HSM solution, with "near online" access capabilities of data on tapes, and apparently perfectly suitable to our facility. What we quickly realized was that in practice it could take anywhere from a few minutes to many hours to either migrate or recall HSM files, depending on their size and their numbers. Most importantly, HSM recall operations run at the highest priority by design on the TSM system, and therefore we observed contention for access of the tape drives while regular backups were taking place. That made the "near online" idea of a "tape based" file system extension impractical from our perspective, at least in the sense that users should not expect to access HSM files from running jobs directly.

Nevertheless we didn't give up, and decided to further push the "archive" part of the idea, and we came up with a reasonable compromise: we devised a 2-step migration system and deployed a 15 TB transient file system called `/repository`, accessible from the datamover servers. On step 1, users may relocate data as required from the active file systems to `/repository` in a number of ways, such as copy, move, tar or rsync. "Transient" refers to the fact that `/repository` works like a "Black Hole": on step 2, `/repository` is constantly being emptied in the background (by the HSM daemons), even while users relocate data in from other file systems. What is left behind is the directory tree with the HSM stub files and the metadata associated with them. In this scenario, on step 2 users also have the option to manually migrate or recall files between `/repository` and the "tape system" with simple commands such as 'dsmmigrate' or 'dsmrecall'.

We also found that performance of this 2-step process was as much a function of the number of files as of the amount of data. Until we had worked diligently with our users to substantially reduce the number of small files on the active file systems we would not be able to proceed. Ideally we would like to see average file sizes greater than 100MB in `/repository`, and recommended that users stage their data ahead of time in large tar-balls.

At this point we still see the contention for tape drives, since we only have 4 units, and are considering a new tape library just for HSM purposes.

8. Large Hadron Collider Computing Grid and the ATLAS Experiment

The SciNet infrastructure provides the largest of the five ATLAS Tier-2 Analysis Centres in Canada (see reference [37] for a more complete description of the Worldwide Large Hadron Collider Computing Grid (WLCG) and the ATLAS-Canada Computing Model). The resources provided at SciNet equal those provided by WestGrid, which has its Tier-2 centres distributed across the University of Victoria, Simon Frasier University, and the University of Alberta. The sheer size of the SciNet Tier-2 centre and the design of the SciNet systems provide some unique challenges to deploying the WLCG middleware and the ATLAS software stack and operating the analysis centre successfully. There is a continuing effort to improve the efficiency of the site in addition to increasing its size to meet the commitments to the experiment. The support is provided by two dedicated FTE's.

The original ATLAS request for computational hours only represented a few percent of the total compute capacity of the ethernet-portion of the GPC. Therefore the grid middleware and ATLAS software could not completely dictate the design for the SciNet infrastructure. However, there was already extensive experience at the University of Toronto with running a Tier-2 centre on a dedicated high-energy physics compute cluster (with 220 dual single-core nodes and 30 TB of storage) which informed some of the RFP requirements and original design choices. We describe some of the original approaches attempted and various mitigation scenarios that were developed as experience was gained at the much larger scale of SciNet. The requested storage infrastructure for ATLAS always represented a significant fraction of the planned SciNet deployment and the

IBM response utilizing their DCS9000 series storage, as described in Section 4, provided a common solution for ATLAS and for the rest of the SciNet users.

8.1. Storage Resource Management and dCache

The components making up the storage element (SE) and storage resource management (SRM) layers were dictated by the previous experience with the dCache filesystem [38] and the fact this was the infrastructure supported by the Canadian ATLAS Tier-1 centre based at TRIUMF [39]. As TRIUMF had already successfully deployed and was running a DDN-based solution provided by IBM, the SciNet ATLAS storage infrastructure offered in the RFP response from IBM was essentially modelled on this solution[3]. Deploying dCache on top of GPFS was not considered at the time as this presented additional layers of complexity which were unnecessary. Given the initial problems with stabilizing the GPFS operation with thousands of nodes accessing the storage via Ethernet, as described in Section 7, it would likely have proven disastrous for the first few months of the SciNet ATLAS Analysis centre. Some thought was given during the RFP process to using the STORM [40] system on top of GPFS (or parallel NFS), but this was always considered as a phase II approach rather than for initial deployment. Again, in hindsight, the data storage solution has been the most successful and stable component of the ATLAS components. The DDN platform with RAID6 deployed over the disk drawers in a silo and verify-on-read and write has proved very stable and almost maintenance free from the dCache perspective with no detected data loss or corruption in the first year of operation. There are not many single points of failure in the chain, although there have been enforced downtimes associated with single DDN controller failures to ensure that there was no chance of data corruption.

The current system provides about 530 TB for ATLAS and is scalable to $\sim$1 PB. Figure 10 shows the growth in the data stored at the SciNet ATLAS Tier-2 data centre since going online. About 5-7 TB of data can be moved a day through the 1 Gbps lightpath to TRIUMF (see Section 8.5).

8.2. WLCG Middleware and ATLAS Software

In the initial deployment of the Tier-2 centre in 2009, the logical choice was to install the grid middleware and ATLAS software on the GPFS filesystem designated `/project`. After much testing in the late fall and early winter it was proven that the performance on GPFS, particularly for the ATLAS software area, was abysmal. This was essentially due to the fact that an ATLAS software release (of which there are dozens) consists of a few GB of data but 100k-1M files, many of which are very small (few kB). Software installation jobs, that are initiated centrally from an ATLAS centralized server at INFN (Italy), would take 2-3 times longer to run at SciNet than at other Canadian Tier-2 sites on similar nodes. Running large numbers of ATLAS analysis jobs also demonstrated that read performance would not scale as many files would be kept open during processing and small amounts of reads would occur throughout the data processing. Increasing the GPFS page pool size, as discussed later, did improve matters, but not dramatically.

A major drawback of the centralized ATLAS software deployment system is that installation of the complete set of releases can take several weeks to accomplish, especially during the initial phase of operation where not all services are robust or hardened. ATLAS software is dependent on `pacman` [41] which is extraordinarily sensitive to the absolute pathname. This installation

[3] The dCache storage is provided by four x3650 dual-cpu 8 GB RAM 'pool' nodes that connect on the backend via a fibre-channel switch to the DDN controllers and connect to clients via the SciNet core Myricom Myri-10G switch. The dCache administration level is provided by four x3550 single-cpu nodes that provide the databases for the filesystem and other services plus an additional four x3550 single-cpu nodes that provide the 'door' services for external transfers via protocols such as GridFTP and GSIDCAP

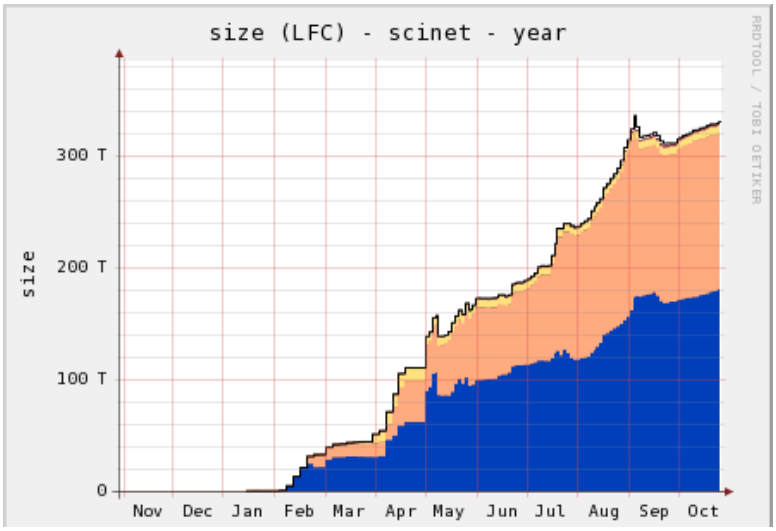

Figure 10. Data storage growth at the SciNet ATLAS Tier-2 Data Centre. Tan represents the event summary data and blue represents the reduced analysis data formats. Older reprocessed datasets are deleted when newer data becomes available.

process was repeated a couple of times in the first few months of operations, once as the paths for the software were changed after deployment had begun and disk configurations were changed and when a second compute element was put online to load balance the job management from the grid.

In early 2010, an NFS server was installed on one of the dCache pool nodes, utilizing one 16 TB tier from the DDN storage as the backend storage. This server proved to scale extremely well, up to 1000's of simultaneous Monte-Carlo simulation jobs during a large production run during the summer. Since then, the DDN tier has been replaced with a RAID0 array constructed from four 146 GB SAS drives. Testing is underway as well on a RAID0 with four 250 GB SATA drives. Very good performance is seen as the NFS caching on the server side scales very well for the hundreds of clients reading essentially the same files as compared to the caching that needed to be done on the client side on individual nodes when GPFS is used.

As discussed in Section 7.4, even backing up the ATLAS software on GPFS was an issue due to the millions of files which created long traversal times for the backup software. Instead, backups are made on NFS servers and tarballs are copied to GPFS areas for storage to tape for long term archiving.

8.3. ATLAS HEP Compute Nodes and GPFS

As discussed in Section 7.1, the GPC compute nodes are diskless and are managed by xCAT and Moab and boot via PXE. As there is only 16 GB RAM per node, keeping the OS and software layers in the deployed image to a bare minimum was deemed essential to efficient use of the systems. The original image deployed under CentOS 5.3 was about 275 MB and has stayed roughly at this size, with modules provided from the GPFS `/project` space adding the necessary software packages and compilers for most SciNet user codes and applications.

It was always understood that the HEP applications, in particular the grid and ATLAS codes, would require substantially more in their compute image than the "standard" vanilla compute image based on CentOS 5.x. In particular many standard rpms usually found on workstations are expected by the ATLAS pilot and Athena code distribution [42], in addition to some legacy rpms to support the 32-bit builds that were still being used in early 2010. A HEP image was designed

based on the standard SciNet compute image, with additional packages originally adding about 300 MB to the footprint. The hope was that these images might converge in time, but a number of factors make this unlikely. The number of required packages for successful operation of grid jobs, though finite, has caused the image to grow to about 1 GB. Another major difference in the requirements is that the HEP jobs need external network connectivity to operate with the Grid and with ATLAS job and data management layers. This connectivity operates through a NAT node which also serves as the main data mover node for SciNet (via 10 Gbps Ethernet). To date, no other user group outside of HEP has needed or requested external connectivity for the compute nodes.

An additional requirement from ATLAS that has yet to be implemented fully at SciNet is that each running job should have access to 2 GB of physical RAM and 3 GB of virtual memory, usually provided by swap. This requirement for ATLAS sites was only requested by the experiment in the last year, so was not in the original design or RFP. Various schemes are being tested to see if providing swap space to the kernel over either GPFS or on an NFS server is practical. The concern with any swap space provided over a networked file systems is that the node performance could be severely compromised, so this needs to be studied before going into large scale deployment. To date, very few jobs have been lost due to lack of memory on the node, but this has occurred. The jobs that tend to have these large footprints are user analysis jobs, rather than the more stable and better controlled simulation production jobs. Some of these failures have been critical as the node would run out of memory and start killing essential services e.g. the GPFS daemon which has a large footprint. Various mitigation techniques have been tried with limited success in telling the OS and Out-Of-Memory killer to sacrifice user applications first, but this does not always help.

One technique in the short term that could help if the available memory becomes a major issue is to limit the number of concurrent jobs on a node. Moab can manage the available memory as a resource to allow nodes to be underfilled. This is not desirable long term or for all jobs as this would be an inefficient use of the available cores on a node, so some tailoring based on which jobs have the large memory footprint would be useful. We have torque-submit helper scripts that can request the necessary resources on the fly, but there is no current mechanism to predict which jobs from the ATLAS job schedulers have these requirements, so this is a manual process at the moment.

Another requirement from the experiment that has doubled in the last year is that ATLAS now requests the availability of 20 GB of scratch disk space per running job. This assumes the potential for up to 14 GB for input files, 5 GB for output and 1 GB for logs etc. The SciNet GPFS `/scratch` space has been used to provide the pseudo-user account home and scratch directories, but not without occasional hiccups. As discussed in Section 7.4, too many files or subdirectories in a directory can prove very inefficient; even simple operations such as listing the files in a directory can sometimes hang for many minutes. After a few months of operation, the `/scratch` space was split off from the `/home` areas for the pseudo-user accounts under which the grid jobs run. Each node now manages its own subdirectory for `/scratch`, the intent being the GPFS file token management for a particular area is limited to a single node. This approach seems to be effective in limiting some of the large wait times that were occasionally seen.

Another approach being investigated is to provide the local scratch for a node with a loopback device, i.e. a large block file is created on GPFS and then mounted on the node with an ext-2 or ext-3 file system built on top of the loopback device. The hope here is to present GPFS with a single file and space-token to manage and all other file operations would be kept local to the node. Given the very variable job load and file access patterns that continually run at SciNet, making definitive statements about improvements or lack thereof from configuration changes takes time to investigate. Some of these approaches were attempted in the early days of the ATLAS deployment on the cluster, but providing space via iSCSI targets or loopback devices did

not prove helpful to throughput performance. However, as the SciNet systems' usage has grown, other issues have emerged with the large number of clients in the GPFS cluster, so minimizing the ATLAS contribution to the pounding of the file systems in a desirable goal.

One major improvement that was discovered in the early days of running while trying to scale the ATLAS athena code to hundreds of concurrent processes was the GPFS pagepool size. To limit the use of physical RAM on a node, this parameter, which controls the caching available to the local GPFS daemon, was originally set to 256 MB in the compute image during the cluster deployment. Extensive testing and detailed measurements of up to 8 analysis jobs running on a node demonstrated that long run-times were being generated by occasional pauses in the GPFS file system feeding data to running processes. These pauses could last seconds to even minutes occasionally. Increasing the pagepool size steadily from 256 MB to 8096 MB demonstrated that the "sweet" spot for ATLAS applications (at least in this test) was around 1 GB and the gains were marginal for anything larger. This large efficiency gain was shown to hold with many nodes running concurrently, though there still is the expected performance impact to having hundreds of jobs simultaneously accessing the same GPFS file system. For a while, the HEP image was deployed with 1 GB and the generic SciNet image was run with 512 MB, but as updating the pagepool size dynamically meant transmitting this information to all other 4,000 nodes, this proved inefficient for node startup times and general GPFS stability, monitoring and performance. All images now use the 1 GB pagepool size.

8.4. ATLAS Job Scheduling

From the outset, SciNet planned to schedule both the GPC and TCS via node and not by core. Grid jobs however need to be scheduled by core, so special routing queues were set up in Moab and Torque to accept serial jobs scheduled through the grid globus job managers. There were various issues that needed to be resolved to ensure these were used efficiently, including some debugging of the Moab software that could cause occasional overprovisioning due to a race condition.

ATLAS requests 48 hours of walltime on modern CPUs for job completion. This synchronizes well with the SciNet scheme as no user job gets more than 48 hours of wallclock time. Since green provisioning has been turned off for the last few months, scheduling these is not a major issue, as these serial jobs only go to the serial queue which can only run on the nodes provisioned with the HEP image. However, we anticipate that green running may be desirable in the future, and tailoring the request time of the grid jobs to be more commensurate with the actual running time would benefit them, as Moab can back-fill schedule node usage much better during the transition times when nodes are being switched from one flavour of image to another. The semantics for providing these job times do not exist in the current WLCG setup; the only way to implement something on this scale currently would be to advertise different length queues to the grid, but no job-matching in the work management schedulers makes use of this information to the authors' knowledge.

We cap the total number of concurrent jobs to protect the storage servers and to meet our commitments to the ATLAS collaboration. These caps have been extended on occasion to encompass special processing requests. In particular, a very large production of simulated data[4] was run over a few weeks during the spring for the jet energy scale calibrations and systematic error studies that were vital in getting some of the first data published from the LHC which set new limits for the production of new particles [27–29]. During this production we ran up to 4,000 ATLAS jobs simultaneously.

To accommodate the large number of jobs, we quickly realized that a single compute element was insufficient to handle the job load from the grid without introducing a bottleneck, especially

[4] Over 3 million events were generated in 60,000+ jobs consuming about 360,000 cpu-hours.

as there tended to be slightly bizarre interactions between the local schedulers and the factories producing the pilot jobs that were sent to the site[5]. We deployed a second compute element which took a little time to configure as both compute elements were scheduling jobs to the same pseudo-user accounts and to the same compute nodes and had to share the same "view" of the universe. Most other large sites that have dedicated resources tend to split their clusters and queues amongst their compute elements eliminating some of the possible confusions and configuration issues. This was not an option at SciNet as we needed to be able to share and steal as many free slots as we could to accomplish the production quickly to meet the deadline of a workshop. This dual compute element configuration has proved useful long term, giving us a level of redundancy. To date, the SciNet Tier-2 has run over 700,000 ATLAS jobs utilizing about 3.5 million cpu-hours.

As mentioned earlier, green provisioning proved to generate a load on GPFS that was deemed unacceptable. In addition, transitioning between the HEP and standard image via dynamic provisioning was contributing to some of the issues and has since been disabled. The length of the job queues from the ATLAS work load management systems for both the simulation and user analysis schedulers does vary frequently albeit with fairly long periodicity. Without dynamic provisioning, daily manual checks are required to ensure that a suitable number of HEP images are deployed and are not sitting idle. Measuring the local idle job queue is not sufficient as the pilot factories are designed to always keep a certain number of jobs queued even if they are not destined to receive a payload.

8.5. Optical lightpath to TRIUMF

It was decided during the initial RFP phases and deployment of the data centre that the simplest wide-area network configuration was to utilize the University of Toronto core network. SciNet appears as just another department in the `utoronto.ca` domain. To accommodate a 1 Gbps point-to-point lightpath connection from the datacentre to TRIUMF, VLAN tagging needed to be turned on to allow routing of traffic on the internal Layer-2 network from the LCG nodes to a separate VLAN which connected to the CANARIE[6]-provided fibre that is hosted on the SciNet edge switch before it hits the main campus router. This was fairly simple to accomplish. Upgrading this link to a full 10 Gbps in the future may require lighting up further wavelengths and providing additional ports on the main Force 10 edge switch at the data centre.

9. Issues of Scale

A variety of other issues arose over SciNet's first year of operation that were caused by system software simply not scaling up to the size of the GPC.

Very few pieces of software, even HPC-specific tools, come "out of the box" ready and tuned to be run on almost 4000 nodes. One of our first experiences with this was our scheduler, where compiled-in limits for maximum number of jobs were far too small, 32-bit integers overflowed, and communications between the resource manager and the nodes was happening far too frequently on a 4000 node system, flooding the network. However, we had an excellent working relationship with our scheduler vendor (Cluster Resources, now Adaptive Computing) and patches were made for us nearly immediately, and fed back into the shipping product.

Similarly, MPI libraries—even commercial MPI libraries—were completely unable to launch several-thousand-task jobs (a) reliably, and (b) in a reasonable amount of time, sometimes taking literally hours before mpirun and/or `MPI_Init()` would complete. OpenMPI, which had excellent Torque/PBS integration, did somewhat better in this respect using the `tm` interface

[5] ATLAS jobs are run via pilot jobs that arrive at the site and ensure a healthy environment before pulling a payload with the real job from centralized PanDA databases [43]. If no payload is found, the pilots self-terminate after a couple of minutes.

[6] Canada's Advanced Research and Innovation Network [44]

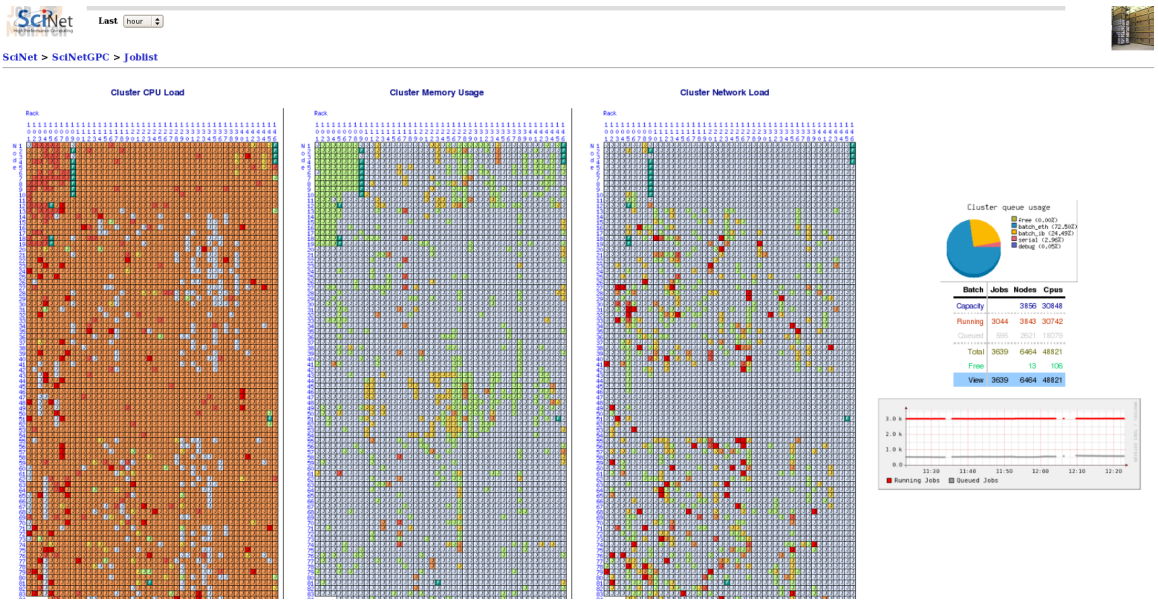

Figure 11. An experimental GPC-wide dashboard using Ganglia and Job-Monarch.

to start jobs, but it was still unusable for thousands of tasks. It was only with the advent of the hydra process manager in MPICH2, which is now being introduced into the newest version of IntelMPI, that we can routinely start jobs of this scale. MPI issues remain in some cases – running large, all-to-all dominated jobs (e.g., FFTs) on Infiniband, something like memory pinning requires too much memory and jobs can fail; running the newest OFED stack ameliorates this but does not eliminate the problem.

On any system, monitoring is crucial, and with the shared file systems monitoring the behaviour of user jobs is particularly urgent. But no dashboard-type monitoring systems run easily out of the box on our large system, as the monitoring has to be very light-weight and have a very small network footprint. Our sysadmin team has developed an experimental GPC dashboard (Fig 11) based on logging infrastructure developed in house, which is already enormously helpful in giving us an 'at a glance' look at the behaviour of the entire system and the jobs on it, allowing us to find issues before they become problems.

I/O continues to come up as an issue in sometimes surprising ways. Beyond continuously working to improve users' jobs, both for their sake and for those of other users, smaller surprises emerge; for instance, our original file system (`/scinet`) containing frequently used libraries (fft, mpi, etc) and applications, did not require a great deal of space, so we originally had it on a very small filesystem containing only a few disks; however, because it was being repeatedly accessed by almost all jobs on almost all nodes, it had to be placed on a much larger filesystem so that it could be striped over many spindles to give us the bandwidth and IOPs we needed, or else it would cause a bottleneck for jobs starting up.

10. Conclusion

To borrow a line from complex systems research, "more is different". Building, operating, and running code on a system of this scale is very different than dealing with smaller systems. By maximizing flexibility, and having a small agile team of large-scale computing specialists working closely with vendors, the SciNet experience has been a success and continues to be a challenge and a learning opportunity.

The SciNet systems have enabled computational research at a scale unprecedented within Canada. From January 2010 to the end of October, researchers have used approximately 200 million CPU hours on SciNet systems to perform simulations and data analysis, pushing forward the fields of biomedicine, climate research, astrophysics, aerospace, chemistry, high energy physics, and many other research areas. We hope the lessons contained within this document help the designers and maintainers of future systems reach this level of productivity even faster.

Acknowledgments
SciNet is funded by: the Canada Foundation for Innovation under the auspices of Compute Canada; the Government of Ontario; Ontario Research Fund - Research Excellence; and the University of Toronto.

Appendix A. Temperature data for Toronto

Table A1 gives temperature data for Toronto.

Table A1. ASHRAE Toronto weather bin data.

Average bin temp (°F)	OAW (gr/lb)	OAH (BTU/lb)	Bins per time period			Total Bin Hours
			12am-8am	8am-4pm	4pm-12am	
-32			0	0	0	0
-27			0	0	0	0
-22			0	0	0	0
-18			0	0	0	0
-13			0	0	0	0
-8	-9.0	-3.3	1	0	0	1
-3	-7.0	-1.8	6	0	0	6
2	1.0	0.6	39	13	15	67
7	7.0	2.7	57	29	52	138
12	11.0	4.6	110	83	81	274
17	16.0	6.5	151	84	131	366
22	21.0	8.5	177	151	135	463
27	26.0	10.5	257	224	226	707
32	30.0	12.3	330	213	246	789
37	35.0	14.3	247	234	260	741
42	39.0	16.1	202	185	242	629
47	43.0	17.9	226	212	186	624
52	49.0	20.1	286	202	236	724
57	53.0	21.9	310	180	259	749
62	57.0	23.7	219	201	249	669
67	61.0	25.6	184	217	217	618
72	65.0	27.4	98	277	231	606
77	67.0	29.0	19	238	95	352
82	70.0	30.7	1	122	45	168
87	73.0	32.3	0	45	12	57
92	77.0	34.2	0	10	2	12
Total						8760

Appendix B. SciNet by the Numbers

Table B1 gives some interesting (to us!) statistics about our systems.

Table B1. SciNet by the numbers.

Number of jobs run so far	3,407,060
File systems (usable size)	
home	14 TB
project	364 TB
scratch	466 TB
dCache	536 TB
Peak number of files	
`/scratch`	320 million
`/project`	110 million
Current number of files	
`/scratch`	40 million
`/project`	20 million
`/scratch` space used	
Peak	461 TB
Typical	410 TB $\pm$ 23
`/scratch` Purging cycle	not accessed in 90 days
average # of files/directory	100
current file type distribution	
binary	90%
ascii	9%
generic	1%
Support emails received	15,640
RAM DIMMs on GPC	$\sim$ 30,000
DIMM failures	150 in the last six months
Slots in tape library	396
Backed up data	294 TB over 310 LTO4 tapes

References

[1] Lovelace G, Scheel M and Szilagyi B 2010 *Preprint arXiv:1010.2777*

[2] Alvarez M and Abel T 2010 *Preprint arXiv:1003.6132*

[3] Pfrommer C and Dursi L 2010 *Nature Physics*

[4] Duez M, Foucart F, Kidder L, Ott C and Teukolsky S 2010 *Classical and Quantum Gravity* **27** 114106

[5] Pang B, Pen U, Perrone M, Treibig J, Wellein G, Hager G, Altisen K, Liu Y, Moy M, Wang Y *et al. Architecture* **163** 643

[6] Chluba J, Vasil G and Dursi L *Monthly Notices of the Royal Astronomical Society*

[7] Fowler J, Acquaviva V, Ade P, Aguirre P, Amiri M, Appel J, Barrientos L, Battistelli E, Bond J, Brown B *et al.* 2010 *The Astrophysical Journal* **722** 1148

[8] Foucart F, Duez M, Kidder L and Teukolsky S 2010 *Preprint arXiv:1007.4203*

[9] Battaglia N, Bond J R, Pfrommer C, Sievers J L and Sijacki D 2010 *Preprint arXiv:1003.4256*

[10] Marriage T A *et al.* 2010 *Preprint arXiv:1010.1065*

[11] Sehgal N *et al.* 2010 *Preprint arXiv:1010.1025*

[12] Dunkley J *et al.* 2010 *Preprint arXiv:1009.0866*

[13] Das S *et al.* 2010 *Preprint arXiv:1009.0847*

[14] Hajian A *et al.* 2010 *Preprint arXiv:1009.0777*

[15] Marriage T A *et al.* 2010 *Preprint arXiv:1007.5256*

[16] Menanteau F *et al.* 2010 *Preprint arXiv:1006.5126*

[17] Chluba J and Thomas R 2010 *Preprint arXiv:1010.3631*

[18] Ivan L, Northrup S A, Groth C P T and De Sterck H 2010 *Proceedings of the 18th Annual Conference of the CFD Society of Canada*

[19] Saffaripour M, Zabeti P, Dworkin S, Zhang Q, Guo H, Liu F, Smallwood G and Thomson M 2010 *Proceedings of the Combustion Institute*

[20] Northrup S A and Groth C P T 2010 *Proceedings of the 18th Annual Conference of the CFD Society of Canada*

[21] Jha P K and Groth C P T 2010 *Proceedings of the Combustion Institute/Canadian Section Spring Technical Meeting*

[22] Groth C P T, Lin W, Hernández-Pérez F E and Gülder Ö L 2010 *Proc. of the Combustion Inst./Canadian Section Spring Technical Meeting*

[23] Jha P K and Groth C P T 2010 *Proc. of the 18th Annual Conf. of the CFD Society of Canada*

[24] Hernández-Pérez F E, Yuen F T C, Groth C P T and Gülder Ö L 2010 *Proc. of the Combustion Inst.* vol 33

[25] Charest M R J, Joo H I, Gülder Ö L and Groth C P T 2010 *Proc. of the Combustion Inst.* vol 33

[26] Zhou Y and Segal D 2010 *The Journal of Chemical Physics* **133** 094101

[27] Aad G *et al.* (ATLAS) 2010 *Phys. Rev. Lett.* **105** 161801 (*Preprint* `1008.2461`)

[28] Aad G *et al.* (ATLAS) 2010 *Preprint arXiv:1009.5069*

[29] Aad G *et al.* (ATLAS) 2010 *Preprint arXiv:1009.5908*

[30] Aad G *et al.* (ATLAS) 2010 *Preprint arXiv:1010.2130*

[31] Dursi L J, Bohlender D, Wadsley J and Kavelaars J J 2010 *White Papers for the 2010 Canadian Astronomy Long Range Plan* URL http://www.casca.ca/lrp2010/Docs/LRPReports/CDandN_WP.pdf

[32] URL http://www-03.ibm.com/systems/software/gpfs/

[33] URL http://www.clusterresources.com/products/moab-cluster-suite/workload-manager.php

[34] URL http://xcat.sourceforge.net/

[35] IBM 2009 IBM Rear Door Heat eXchanger for the iDataPlex Rack Installation and Maintenance Guide URL http://publib.boulder.ibm.com/infocenter/idataplx/documentation/topic/com.ibm.idataplex.doc/dg1dimst.pdf

[36] IBM 2007 IBM Canada's innovative water cooling project recognized for innovation and energy efficiency URL http://www.ibm.com/ibm/environment/news/enerstat_2007.shtml

[37] ATLAS Canada Collaboration 2006 URL http://atlas-canada.web.cern.ch/atlas-canada/documents/AtCanComp-PublicV1.pdf

[38] URL http://www.dcache.org

[39] URL http://atlas.triumf.ca

[40] URL http://storm.forge.cnaf.infn.it

[41] URL http://www.archlinux.org/pacman

[42] 2005 *The Athena Control framework in production, new developments and lessons learned*

[43] Nilsson P *et al.* (ATLAS) 2008 *PoS* **ACAT08** 027

[44] URL http://www.canarie.ca